UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/880,024 | 06/14/2001 | Serban I. Gavrila | 068398-0106 | 5709 |

23838 7590 12/02/2004

KENYON & KENYON
1500 K STREET, N.W., SUITE 700
WASHINGTON, DC 20005

| EXAMINER | |
|---|---|
| PARTHASARATHY, PRAMILA | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2136 | |

DATE MAILED: 12/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/880,024 | GAVRILA ET AL. | |
| | Examiner | Art Unit | |
| | Pramila Parthasarathy | 2136 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *03 April 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

**1.** This action is in response to the communication filed on 04/03/2003. Claims 1 – 42 were received for consideration. No preliminary amendments to the specification were filed. Claims 1 – 42 are currently being considered.

### *Claim Rejections - 35 USC § 102*

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

**2.** Claims 1 – 13, 15 – 34 and 36 – 42 are rejected under 35 U.S.C. 102(e) as being anticipated by Helland et al. (U.S. Patent Number 6,014,666).

Regarding Claim 1, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment comprises a role-based access control system that includes a directed acyclic graph

representing role-membership inheritance relationships and a directed acyclic graph representing role-permission inheritance relationships, said method comprising:

associating each role with the set of abstract objects accessible to the said role, said association requiring neither redundant storage and maintenance of permissions nor exhaustive system searches (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38).

Claim 22. A computer program product containing computer readable code for causing a machine to perform the following method steps:

automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment comprises a role-based access control system that includes a directed acyclic graph representing role-membership inheritance relationships and a directed acyclic graph representing role-permission inheritance relationships;

association of each role with the set of abstract objects accessible to the said role, said association requiring neither redundant storage and maintenance of permissions nor exhaustive system searches (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38).

Claims 2 and 23 are rejected applied as above in rejecting Claim 1 and 22. Furthermore, Helland teaches and describes a method for the automatic distribution,

review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

defining and managing the abstract permissions of a role on abstract objects; and finding, retrieving, and displaying abstract permissions of a role on abstract objects (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

adding an abstract object to the set of abstract objects associated with a role whenever said abstract object becomes accessible to said role (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

deleting an abstract object from the set of abstract objects associated with a role whenever said abstract object becomes inaccessible to said role (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64).

Claims 3 and 24 are rejected applied as above in rejecting Claim 2 and 23. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

creating, finding, retrieving, displaying, and deleting instances of a role on a host computer or set of host computers, using group nesting and a directed acyclic graph of role-membership inheritance (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 22 lines 14 – 40); and

creating finding, retrieving, displaying, and deleting object instances of abstract objects on a host computer or set of host computers (Summary; Column 15 lines 24 –

57; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64); and

registering objects as instances of abstract objects on a host computer or set of host computers (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

deriving permissions of a role instance on object instances from the abstract permissions of said role on said abstract objects (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

registering permissions on objects as instances of abstract permissions on abstract objects on a host computer or set of host computers (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 22 lines 14 – 40); and

finding, retrieving, and displaying the permissions derived from abstract permissions defined on abstract objects (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64).

Claims 4 and 25 are rejected applied as above in rejecting Claim 1 and 22. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising the steps of:

creating an instance of a RBAC user on a set of host computers, said user instance being called global with respect to said set of host computers(Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64); and

creating an instance of a RRAC user on a host computer, said user instance being called local with respect to said host computer, unless said host computer is used to control a set of host computers, in which case the instance is called global with respect to said set of host computers (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 19 lines 59 – 64); and

creating a role instance on a set of host computers, said role instance being called global with respect to said set of host computers (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64); and

creating a role instance on a host computer, said role instance being called local with respect to said host computer, unless said host computer is used to control a set of host computers, in which case one can select whether the instance will be local with respect to said host computer, or global with respect to said set of host computers (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64); and

including a local user instance in a local role instance, if said user is assigned to said role, and both said instances were derived on the same host computer (Summary; Column 17 lines 21 – 54 and Column 19 lines 59 – 64); and

including a global user instance in a local role instance, if said user is assigned to said role, and said local role instance was derived on a host computer included in the set of host computers used to derive said global user instance (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64); and

including the global user instance in a global role instance, if said user is assigned to said role, and both said instances were derived on the same set of host computers (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64); and

including the members of a local instance of a first role in a local instance of a second role, if the second role inherits the membership of the first role, and both said instances were derived on the same host computer (Summary; Column 17 lines 21 – 54 and Column 19 lines 59 – 64); and

including the global instance of a first role as a member of a local instance of a second role, if the second role inherits the membership of the first role, and said local instance was derived on a host computer included in the set of host computers used to derive said global instance (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64); and

including the members of a global instance of a first role in a global instance of a second role, if the second role inherits the membership of the first role, and both said instances were derived on the same set of host computers (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64).

Claims 5 and 26 are rejected applied as above in rejecting Claim 3 and 24. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

computing, displaying, reviewing, and listing the permissions of any role to abstract objects (Summary; Column 15 lines 24 – 57; and Column 17 lines 21 – 54); and

computing, displaying, reviewing, and listing the permissions of any role to object instances (Summary; Column 15 lines 24 – 57; and Column 17 lines 21 – 54); and

computing, displaying, reviewing, and listing the permissions of any role instance to object instances(Summary; Column 15 lines 24 – 57; and Column 17 lines 21 – 54).

Claims 8 and 29 are rejected applied as above in rejecting Claim 3 and 24. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

automatic distribution of permissions on object instances to role instances whenever new permission-inheritance relations are established among roles (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatic distribution of permissions on object instances to role instances whenever new roles are added to the directed acyclic graph (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatic distribution of permissions on object instances to role instances whenever a new role instance is created for a role on a host computer or set of host computers (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatic distribution of permissions on object instances to role instances whenever a new object instance is created for an abstract object on a host computer or set of host computers (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatic distribution of permissions on object instances to role instances whenever a new permission is granted to a role (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38).

Claims 9 and 30 are rejected applied as above in rejecting Claim 3 and 24. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

automatic revocation and recalculation of permissions on object instances for role instances whenever permission-inheritance relations among roles are removed (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatic revocation and recalculation of permissions on object instances for role instances whenever roles are removed (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatic revocation and recalculation of permissions on object instances for roles instances whenever an abstract object is removed (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatic revocation and recalculation of permissions on object instances for role instances whenever a permission is revoked from a role (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38).

Claims 10 and 31 are rejected applied as above in rejecting Claim 3 and 24. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

scaleable, automatic, distribution, revocation, and recalculation of permissions of role instances to object instances that support efficient access authorization (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 19 lines 59 – 64).

Claims 6 and 27 are rejected applied as above in rejecting Claim 5 and 26. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

determining whether two or more roles share permissions on any abstract objects (Summary and Column 17 lines 21 – 54); and

determining whether two or more roles share permissions on any object instances (Summary and Column 17 lines 21 – 54); and

determining whether two or more role instances share permissions on any object instances (Summary and Column 17 lines 21 – 54); and

implementing and testing any policy that is satisfied by the determination of whether two or more roles share permissions to abstract objects (Summary; Column 17 lines 21 – 54 and Column 22 lines 14 - 40); and

implementing and testing any policy that is satisfied by the determination of whether two or more roles share permissions to object instances (Summary; Column 17 lines 21 – 54 and Column 22 lines 14 - 40); and

implementing and testing any policy that is satisfied by the determination of whether two or more role instances share permissions to object instances (Summary; Column 17 lines 21 – 54 and Column 22 lines 14 - 40).

Claims 11 and 32 are rejected applied as above in rejecting Claim 10 and 31. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

adding a new permission-inheritance arc to the directed acyclic graph between a first role called inheritor role and a second role called the inherited role whereby the inheritor and all its ascendant roles inherit all the permissions of the inherited role and its descendant roles in the directed acyclic graph (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatically selecting the roles that do not have instances on a host computer or set of host computers from the set comprises the said inherited role and its

descendants in the directed acyclic graph (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 22 lines 14 – 40); and

automatically computing a set of permissions by mapping the abstract permissions of said selected roles on all abstract objects that do have instances on said host computer or set of host computers (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 22 lines 14 – 40); and

automatically granting said computed permissions to the instance of each first encountered role instantiated on said host computer or set of host computers by traversing the directed acyclic graph in the direction opposite to that of the inheritance arcs on any path starting from the inheritor role (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 22 lines 14 – 40).

Claims 15 and 36 are rejected applied as above in rejecting Claim 10 and 31. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

creating an instance of a role on a host computer or set of host computers; and automatically selecting the roles that did not have instances on said host computer or set of host computers prior to the creation of said role instance, wherein the selection is performed from said role and its descendant roles in the directed acyclic graph (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 22 lines 14 – 40); and

automatically computing a set of permissions by mapping the abstract permissions of said selected roles on all abstract objects that do have instances on said host computer or set of host computers (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 22 lines 14 – 40); and

automatically granting said computed permissions to said role instance just created (Summary; Column 17 lines 21 – 54; Column 18 lines 10 – 38 and Column 22 lines 14 – 40).

Claims 16 and 37 are rejected applied as above in rejecting Claim 10 and 31. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

creating an instance of a user on a host computer or set of host computers (Summary; Column 18 lines 10 – 38 and Column 22 lines 14 – 40); and

automatically selecting the roles that did not have instances on said host computer or set of host computers prior to the creation of said user instance, wherein the selection is performed from said user and its descendant roles in the directed acyclic graph (Summary; Column 18 lines 10 – 38 and Column 22 lines 14 – 40); and

automatically computing a set of permissions by mapping the abstract permissions of said selected roles on all abstract objects that do have instances on said host computer or set of host computers (Summary and Column 18 lines 10 – 38); and

automatically ranting said computed permissions to said user instance just created (Summary and Column 18 lines 10 – 38).

Claims 17 and 38 are rejected applied as above in rejecting Claim 10 and 31. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

granting a role an abstract permission to an abstract object that has an instance on a host computer or set of host computers and automatically causing the said role's ascendant roles and users to inherit the said abstract permission (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatically updating the association between the said role and the set of accessible abstract objects (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatically mapping the said abstract permission of said role on said abstract object to a set of permissions for the object instance (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatically granting said set of permissions to the instance of each first encountered role instantiated on said host computer or set of host computers by traversing the directed acyclic graph in the direction opposite to that of the inheritance arcs on any path starting from the role being granted the abstract permission

(Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38).

Claims 18 and 39 are rejected applied as above in rejecting Claim 10 and 31. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, Claim 18. The method of claim 10, further comprising:

instantiating an abstract object on a host computer or set of host computers (Summary and Column 18 lines 10 – 38); and

automatically reading the access control list of the abstract object and computing the set of roles that have abstract permissions to the said abstract object (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

for each role in the said set, automatically mapping the abstract permissions of said role on said abstract object to a set of permissions for the object instance (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatically granting said set of permissions to the instance of each first encountered role instantiated on said host computer or set of host computers by traversing the directed acyclic graph in the direction opposite to that of the inheritance arcs on any path starting from said role (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38).

Claims 19 and 40 are rejected applied as above in rejecting Claim 10 and 31. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising

deleting an abstract object, including the steps: automatically finding and deleting all instances of said abstract object and their access control lists; and automatically reading the access control list of said abstract object and, for each role found in the said access control list, removing the said abstract object from the association between said role and its set of accessible abstract objects; and automatically deleting the said abstract object and its access control list (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54 and Column 18 lines 10 – 38).

Claims 20 and 41 are rejected applied as above in rejecting Claim 10 and 31. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

deriving a directed acyclic graph of roles representing both membership and permission inheritance, abstract objects, and abstract permissions, from the user account, group, and access control list and permission structures of extant operating systems (Summary; Column 15 lines 24 – 57; Column 17 lines 21 – 54; Column 18 lines 10 – 38); and

performing the incremental transition from an extant permission management system to automatic permission management in RBAC (Column 18 line 10 – Column 19 line 64).

Claims 7 and 28 are rejected applied as above in rejecting Claim 6 and 27. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

implementing and testing generalized separation-of-duty policies (Summary and Column 19 line 66 – Column 20 line 55); and

implementing and testing operational separation-of-duty policies (Summary and Column 19 line 66 – Column 20 line 55).

Claims 12 and 33 are rejected applied as above in rejecting Claim 11 and 32. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

removing a permission-inheritance arc from the directed acyclic graph between a first role called inheritor role and a second role called the inherited role (Summary and Column 20 line 22 – Column 21 line 47); and

automatically recalculating permissions and granting said permissions to the instance of each first encountered role instantiated on a host computer or set of host

computers, by traversing the directed acyclic graph in the direction opposite to that of the inheritance arcs on any path starting from the inheritor role (Summary and Column22 line 14 – 65).

Claims 13 and 34 are rejected applied as above in rejecting Claims 11 and 32. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

revoking an abstract permission to an abstract object from a role where said abstract object has an instance on a host computer or set of host computers (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatically updating the association between the said role and the set of accessible abstract objects (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38); and

automatically recalculating permissions and granting said permissions to the instance of each first encountered role instantiated on a host computer or set of host computers, by traversing the directed acyclic graph in the direction opposite to that of the inheritance arcs on any path starting from the said role (Summary; Column 17 lines 21 – 54 and Column 18 lines 10 – 38).

Claims 21 and 42 are rejected applied as above in rejecting Claim 20 and 41. Furthermore, Helland teaches and describes a method for the automatic distribution,

review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment, further comprising:

deriving membership-inheritance and permission-inheritance relationships among the existing user accounts and groups (Summary; Column 17 lines 21 – 54); and

creating roles and assigning selected user accounts and groups to said roles (Summary; Column 17 lines 21 – 54); and

deriving membership-inheritance and permission-inheritance relationships among said roles and obtaining a directed acyclic graph for each type of inheritance relationship (Summary; Column 15 lines 24 – 57 and Column 17 lines 21 – 54); and

transforming the said directed acyclic graphs into a single directed acyclic graph of membership inheritance that preserves the permission of the user accounts defined by permission inheritance (Summary; Column 15 lines 24 – 57 and Column 17 lines 21 – 54).

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

**3.** Claims 14 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Helland et al. (U.S. Patent Number 6,014,666, hereinafter "Helland") in view of Khun (U.S. Patent Number 6,023,765, hereinafter "Khun").

Claims 14 and 35 are rejected applied as above in rejecting Claims 11 and 32. Furthermore, Helland teaches and describes a method for the automatic distribution, review and revocation of user and group permissions to objects through management of role permissions to abstract objects in a computing environment.

Helland does not explicitly teach deleting a role from the directed acyclic graph, further comprising: selecting a role for deletion from the directed acyclic graph; automatically removing the said role from the access control lists of all abstract objects accessible to said role; and automatically deleting the association between said role and all abstract objects accessible to said role; and automatically recalculating permissions and granting said permissions to the instance of each first encountered role instantiated on a host computer or set of host computers, by traversing the directed acyclic graph in the direction opposite to that of the inheritance arcs on any path starting from the any direct ascendant of the selected; and automatically deleting all instances of the selected; and automatically deleting the selected role from the directed acyclic graph (Helland Summary; Column 15 line 24 – Column 19 line 64). However, Kuhn discloses a method to implement RBAC (Role-based access control) wherein security is managed at object level by assigning roles and a particular advantage of RABC is that it allows the access privileges provided to be conveniently reconfigured by deleting one's role

from the directed acyclic graph, further comprising: selecting a role for deletion from the directed acyclic graph; automatically removing the said role from the access control lists of all abstract objects accessible to said role; and automatically deleting the association between said role and all abstract objects accessible to said role (Kuhn Column 2 line 29 – Column 3 line 4 and Column 4 line 26 – Column 5 line 9); and

automatically recalculating permissions and granting said permissions to the instance of each first encountered role instantiated on a host computer or set of host computers, by traversing the directed acyclic graph in the direction opposite to that of the inheritance arcs on any path starting from the any direct ascendant of the selected; and automatically deleting all instances of the selected; and automatically deleting the selected role from the directed acyclic graph (Kuhn Column 10 line 11 – Column 12 line 61).

Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify Kuhn's automatically maintaining permission to the instances of each role into automatically maintaining permission on object instances as taught by Helland. One of ordinary skill in the art would have been motivated to modify Helland by Khun because maintaining permission to the instances of each role is fundamental to RBAC and in any RBAC system, access to an object within a computer system is provided to the members of groups by assigning roles, which gives same privileges to access various objects within the system and a particular advantage of RBAC is that it allows very convenient reconfigurations by deleting a role.

***Conclusion***

**4.** Any inquiry concerning this communication or earlier communications from the examiner should be directed to Pramila Parthasarathy whose telephone number is 571-272-3866. The examiner can normally be reached on 8:00a.m. To 5:00p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on 571-232-3795.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR only. For more information about the PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Pramila Parthasarathy
November 27, 2004.

E. Moise
EMMANUEL L. MOISE
PRIMARY EXAMINER
11/4 2136